khal Documentation

Release 0.9.2

Christan Geier et al.

February 14, 2017

Contents

Khal is a standards based CLI (console) calendar program, able to synchronize with CalDAV servers through vdirsyncer.

Features

(or rather: limitations)

- khal can read and write events/icalendars to vdir, so vdirsyncer can be used to synchronize calendars with a variety of other programs, for example CalDAV servers.

- fast and easy way to add new events

- ikhal (interactive khal) lets you browse and edit calendars and events

- only rudimentary support for creating and editing recursion rules

- you cannot edit the timezones of events

- works with python 3.3+

- khal should run on all major operating systems [1]

---

[1] except for Microsoft Windows

Table of Contents

2.1 Installation

If khal is packaged for your OS/distribution, using your system's standard package manager is probably the easiest way to install khal. khal has been packaged for, among others: Arch Linux (stable and development versions), Debian, Fedora, FreeBSD, Guix, and pkgsrc.

If a package isn't available (or it is outdated) you need to fall back to one of the methods mentioned below.

2.1.1 Install via Python's Package Managers

Since *khal* is written in python, you can use one of the package managers available to install python packages, e.g. *pip*.

You can install the latest released version of *khal* by executing:

```
pip install khal
```

or the latest development version by executing:

```
pip install git+git://github.com/pimutils/khal.git
```

This should also take care of installing all required dependencies.

Otherwise you can always download the latest release from pypi and execute:

```
python setup.py install
```

or better:

```
pip install .
```

in the unpacked distribution folder.

Since version 0.8 *khal* does **only supports python 3**. If you have python 2 and 3 installed in parallel you might need to use *pip3* instead of *pip* and *python3* instead of *python*. In case your operating system cannot deal with python 2 and 3 packages concurrently, we suggest installing *khal* in a virtualenv (e.g. by using virtualenvwrapper or with the help of pipsi) and than starting khal from that virtual environment.

2.1.2 Requirements

khal is written in python and can run on Python 3.3+. It requires a Python with `sqlite3` support enabled (which is usually the case).

If you are installing python via *pip* or from source, be aware that since *khal* indirectly depends on lxml you need to either install it via your system's package manager or have python's libxml2's and libxslt1's headers (included in a separate "development package" on some distributions) installed.

2.2 Configuration

khal reads configuration files in the *ini* syntax, meaning it understands keys separated from values by a **=**, while section and subsection names are enclosed by single or double square brackets (like **[sectionname]** and **[[subsection-name]]**).

2.2.1 Help with initial configuration

If you do not have a configuration file yet, running **khal configure** will launch a small, interactive tool that should help you with initial configuration of **khal**.

2.2.2 Location of configuration file

khal is looking for configuration files in the following places and order: `$XDG_CONFIG_HOME/khal/config` (on most systems this is `~/.config/khal/config`), `~/.khal/khal.conf` and a file called `khal.conf` in the current directory. Alternatively you can specify which configuration file to use with `-c path/to/config` at runtime.

The [calendars] section

The *[calendars]* section is mandatory and must contain at least one subsection. Every subsection must have a unique name (enclosed by two square brackets). Each subsection needs exactly one *path* setting, everything else is optional. Here is a small example:

```
[calendars]

  [[home]]
    path = ~/.calendars/home/
    color = dark green

  [[work]]
    path = ~/.calendars/work/
    readonly = True
```

color

    khal will use this color for coloring this calendar's event. The following color names are supported: *black*, *white*, *brown*, *yellow*, *dark gray*, *dark green*, *dark blue*, *light gray*, *light green*, *light blue*, *dark magenta*, *dark cyan*, *dark red*, *light magenta*, *light cyan*, *light red*. Depending on your terminal emulator's settings, they might look different than what their name implies. In addition to the 16 named colors an index from the 256-color palette or a 24-bit color code can be used, if your terminal supports this. The 256-color palette index is simply a number between 0 and 255. The 24-bit color must be given as #RRGGBB, where RR, GG, BB is the hexadecimal value of the red, green and blue component, respectively. When using a 24-bit color, make sure to enclose the color

value in ' or "! If the color is set to *auto* (the default), khal tries to read the file *color* from this calendar's vdir, if this fails the default_color (see below) is used. If color is set to '', the default_color is always used.

> **type** color
>
> **default** auto

path
> The path to an existing directory where this calendar is saved as a *vdir*. The directory is searched for events or birthdays (see `type`). The path also accepts glob expansion via * or *?* when type is set to discover. This allows for paths such as *~/accounts/*/calendars/*, where the calendars directory contains vdir directories. In addition, *~/calendars/** and *~/calendars/default* are valid paths if there exists a vdir in the *default* directory. (The previous behaviour of recursively searching directories has been replaced with globbing).
>
> > **type** string
> >
> > **default** None

readonly
> setting this to *True*, will keep khal from making any changes to this calendar
>
> > **type** boolean
> >
> > **default** False

type
> Setting the type of this collection (default `calendar`).
>
> If set to `calendar` (the default), this collection will be used as a standard calendar, that is, only files with the `.ics` extension will be considered, all other files are ignored (except for a possible *color* file).
>
> If set to `birthdays` khal will expect a VCARD collection and extract birthdays from those VCARDS, that is only files with `.ics` extension will be considered, all other files will be ignored. `birthdays` also implies `readonly=True`.
>
> If set to `discover`, khal will use globbing to expand this calendar's *path* to (possibly) several paths and use those as individual calendars (this cannot be used with *birthday* collections'). See *Examplary discover usage* for an example.
>
> If an individual calendar vdir has a *color* file, the calendar's color will be set to the one specified in the *color* file, otherwise the color from the *calendars* subsection will be used.
>
> > **type** option, allowed values are *calendar*, *birthdays* and *discover*
> >
> > **default** calendar

The [default] section

Some default values and behaviours are set here.

default_calendar
> The calendar to use if none is specified for some operation (e.g. if adding a new event). If this is not set, such operations require an explicit value.
>
> > **type** string
> >
> > **default** None

default_command
> command to be executed if no command is given when executing khal
>
> > **type** option, allowed values are *calendar*, *list*, *interactive*, *printformats*, *printcalendars*, *printics* and ***

> > **default** calendar

highlight_event_days
> If true, khal will highlight days with events. Options for highlighting are in [highlight_days] section.
>
> > **type** boolean
> >
> > **default** False

print_new
> After adding a new event, what should be printed to standard out? The whole event in text form, the path to where the event is now saved or nothing?
>
> > **type** option, allowed values are *event*, *path* and *False*
> >
> > **default** False

show_all_days
> By default, khal displays only dates with events in *list* or *calendar* view. Setting this to *True* will show all days, even when there is no event scheduled on that day.
>
> > **type** boolean
> >
> > **default** False

timedelta
> Default timedelta to use for reporting, for example in *list*
>
> > **type** timedelta
> >
> > **default** 2d

The [highlight_days] section

When highlight_event_days is enabled, this section specifies how the highlighting/coloring of days is handled.

color
> What color to use when highlighting - explicit color or use calendar color when set to ''
>
> > **type** color
> >
> > **default**

default_color
> Default color for calendars without color - when set to '' it actually disables highlighting for events that should use the default color.
>
> > **type** color
> >
> > **default**

method
> Highlighting method to use - foreground or background
>
> > **type** option, allowed values are *foreground*, *fg*, *background* and *bg*
> >
> > **default** fg

multiple
> How to color days with events from multiple calendars - either explicit color or use calendars' colors when set to ''
>
> > **type** color
> >
> > **default**

The [keybindings] section

keybindings for **ikhal** are set here. You can bind more than one key (combination) to a command by supplying a comma-separated list of keys. For binding key combinations concatenate them keys (with a space in between), e.g. **ctrl n**.

delete
> delete the currently selected event
>
>> **type** list
>>
>> **default** d

down
> move the cursor down (in the calendar browser)
>
>> **type** list
>>
>> **default** down, j

duplicate
> duplicate the currently selected event
>
>> **type** list
>>
>> **default** p

export
> export event as a .ics file
>
>> **type** list
>>
>> **default** e

left
> move the cursor left (in the calendar browser)
>
>> **type** list
>>
>> **default** left, h, backspace

mark
> go into highlight (visual) mode to choose a date range
>
>> **type** list
>>
>> **default** v

new
> create a new event on the selected date
>
>> **type** list
>>
>> **default** n

other
> in highlight mode go to the other end of the highlighted date range
>
>> **type** list
>>
>> **default** o

quit
> quit
>
>> **type** list

> **default** q, Q

right
> move the cursor right (in the calendar browser)
>
> > **type** list
> >
> > **default** right, l, space

save
> save the currently edited event and leave the event editor
>
> > **type** list
> >
> > **default** meta enter

search
> open a text field to start a search for events
>
> > **type** list
> >
> > **default** /

today
> focus the calendar browser on today
>
> > **type** list
> >
> > **default** t

up
> move the cursor up (in the calendar browser)
>
> > **type** list
> >
> > **default** up, k

view
> show details or edit (if details are already shown) the currently selected event
>
> > **type** list
> >
> > **default** enter

The [locale] section

It is mandatory to set (long)date-, time-, and datetimeformat options, all others options in the **[locale]** section are optional and have (sensible) defaults.

dateformat
> khal will display and understand all dates in this format, see *timeformat* for the format
>
> > **type** string
> >
> > **default** %d.%m.

datetimeformat
> khal will display and understand all datetimes in this format, see *timeformat* for the format.
>
> > **type** string
> >
> > **default** %d.%m. %H:%M

default_timezone
> this timezone will be used for new events (when no timezone is specified) and when khal does not understand the timezone specified in the icalendar file. If no timezone is set, the timezone your computer is set to will be used.
>
> > **type** timezone
> >
> > **default** None

firstweekday
> the first day of the week, were Monday is 0 and Sunday is 6
>
> > **type** integer, allowed values are between 0 and 6
> >
> > **default** 0

local_timezone
> khal will show all times in this timezone If no timezone is set, the timezone your computer is set to will be used.
>
> > **type** timezone
> >
> > **default** None

longdateformat
> khal will display and understand all dates in this format, it should contain a year (e.g. *%Y*) see *timeformat* for the format.
>
> > **type** string
> >
> > **default** %d.%m.%Y

longdatetimeformat
> khal will display and understand all datetimes in this format, it should contain a year (e.g. *%Y*) see *timeformat* for the format.
>
> > **type** string
> >
> > **default** %d.%m.%Y %H:%M

timeformat
> khal will display and understand all times in this format.
>
> The formatting string is interpreted as defined by Python's strftime, which is similar to the format specified in `man strftime`.
>
> > **type** string
> >
> > **default** %H:%M

unicode_symbols
> by default khal uses some unicode symbols (as in 'non-ascii') as indicators for things like repeating events, if your font, encoding etc. does not support those symbols, set this to *False* (this will enable ascii based replacements).
>
> > **type** boolean
> >
> > **default** True

weeknumbers
> Enable weeknumbers in *calendar* and *interactive* (ikhal) mode. As those are iso weeknumbers, they only work properly if *firstweekday* is set to 0
>
> > **type** weeknumbers
> >
> > **default** off

The [sqlite] section

path

khal stores its internal caching database here, by default this will be in the *$XDG_DATA_HOME/khal/khal.db* (this will most likely be *~/.local/share/khal/khal.db*).

> **type** string
>
> **default** None

The [view] section

The view section contains configuration options that effect the visual appearance when using ikhal.

agenda_day_format

Specifies how each *day header* is formated.

> **type** string
>
> **default** {bold}{name}, {date-long}{reset}

agenda_event_format

Default formating for events used when the user asks for all events in a given time range, used for **list**, **calendar** and in **interactive** (ikhal). Please note, that any color styling will be ignored in *ikhal*, where events will always be shown in the color of the calendar they belong to. The syntax is the same as for `--format`.

> **type** string
>
> **default** {calendar-color}{start-end-time-style}   {title}{repeat-symbol}{description-separator}{description}{reset}

bold_for_light_color

Whether to use bold text for light colors or not. Non-bold light colors may not work on all terminals but allow using light background colors.

> **type** boolean
>
> **default** True

dynamic_days

Defines the behaviour of ikhal's right column. If *True*, the right column will show events for as many days as fit, moving the cursor through the list will also select the appropriate day in the calendar column on the left. If *False*, only a fixed ([default] timedelta) amount of days' events will be shown, moving through events will not change the focus in the left column.

> **type** boolean
>
> **default** True

event_format

Default formating for events used when the start- and end-date are not clear through context, e.g. for **search**, used almost everywhere but **list** and **calendar**. It is therefore probably a sensible choice to include the start- and end-date. The syntax is the same as for `--format`.

> **type** string
>
> **default** {calendar-color}{start}-{end}   {title}{repeat-symbol}{description-separator}{description}{reset}

event_view_always_visible

Set to true to always show the event view window when looking at the event list

> **type** boolean
>
> **default** False

event_view_weighting
> weighting that is applied to the event view window
>
> > **type** integer
> >
> > **default** 1

frame
> Whether to show a visible frame (with *box drawing* characters) around some (groups of) elements or not. There are currently several different frame options available, that should visually differentiate wether an element is in focus or not. Some of them will probably be removed in future releases of khal, so please try them out and give feedback on which style you prefer (the color of all variants can be defined in the color themes).
>
> > **type** option, allowed values are *False*, *width*, *color* and *top*
> >
> > **default** False

theme
> Choose a color theme for khal.
>
> This is very much work in progress. Help is really welcome! The two currently available color schemes (*dark* and *light*) are defined in *khal/ui/colors.py*, you can either help improve those or create a new one (see below). As ikhal uses urwid, have a look at urwid's documentation for how to set colors and/or at the existing schemes. If you cannot change the color of an element (or have any other problems) please open an issue on *github*.
>
> If you want to create your own color scheme, copy the structure of the existing ones, give it a new and unique name and also add it as an option in *khal/settings/khal.spec* in the section *[default]* of the property *theme*.
>
> > **type** option, allowed values are *dark* and *light*
> >
> > **default** dark

A minimal sample configuration could look like this:

2.2.3 Example

```
[calendars]
[[home]]
path = ~/.calendars/home/

[[work]]
path = ~/.calendars/work/

[locale]
local_timezone= Europe/Berlin
default_timezone= Europe/Berlin
timeformat= %H:%M
dateformat= %d.%m.
longdateformat= %d.%m.%Y
datetimeformat= %d.%m. %H:%M
longdatetimeformat= %d.%m.%Y %H:%M
```

2.2.4 Examplary discover usage

If you have the following directory layout:

```
~/calendars
- work/
- home/
- family/
```

where *work*, *home* and *family* are all different vdirs, each containg one calendar, a matching calendar section could look like this:

```
[[calendars]]
path = ~/calendars/*
type = discover
color = dark green
```

2.2.5 Syncing

To get **khal** working with CalDAV you will first need to setup vdirsyncer. After each start **khal** will automatically check if anything has changed and automatically update its caching db (this may take some time after the initial sync, especially for large calendar collections). Therefore you might want to execute **khal** automatically after syncing with **vdirsyncer** (e.g. via **cron**).

2.3 Usage

Khal offers a set of commands, most importantly **list**, **calendar**, **interactive**, **new**, **printcalendars**, **printformats**, and **search**. See below for a description of what every command does. Calling **khal** without any command will invoke the default command, which can be specified in the configuration file.

2.3.1 Options

khal (without any commands) has some options to print some information about **khal**:

--version
> Prints khal's version number and exits

-h, --help
> Prints a summary of khal's options and commands and then exits

Several options are common to almost all of **khal**'s commands (exceptions are described below):

-v
> Be more verbose (e.g. print debugging information)

-c CONFIGFILE
> Use an alternate configuration file

-a CALENDAR
> Specify a calendar to use (which must be configured in the configuration file), can be used several times. Calendars not specified will be disregarded for this run.

-d CALENDAR
> Specify a calendar which will be disregarded for this run, can be used several times.

--color/-no-color
> **khal** will detect if standard output is not a tty, e.g., you redirect khal's output into a file, and if so remove all highlighting/coloring from its output. Use *--color* if you want to force highlighting/coloring and *--no-color* if you want coloring always removed.

--format FORMAT

For all of khal's commands that print events, the formatting of that event can be specfied with this option. FORMAT is a template string, in which identifiers delimited by curly braces (*{}*) will be expanded to an event's properties. FORMAT supports all formating options offered by python's str.format() (as it is used internally). The available template options are:

title The title of the event.

description The description of the event.

start The start datetime in datetimeformat.

start-long The start datetime in longdatetimeformat.

start-date The start date in dateformat.

start-date-long The start date in longdateformat.

start-time The start time in timeformat.

end The end datetime in datetimeformat.

end-long The end datetime in longdatetimeformat.

end-date The end date in dateformat.

end-date-long The end date in longdateformat.

end-time The end time in timeformat.

repeat-symbol A repeating symbol (loop arrow) if the event is repeating.

description The event description.

description-separator A separator: " :: " that appears when there is a description.

location The event location.

calendar The calendar name.

calendar-color Changes the output color to the calendar's color.

start-style The start time in timeformat OR an appropriate symbol.

to-style A hyphen "-" or nothing such that it appropriatly fits between start-style and end-style.

end-style The end time in timeformat OR an appropriate symbol.

start-end-time-style A concatenation of start-style, to-style, and end-style OR an appropriate symbol.

end-necessary For an allday event this is an empty string unless the end date and start date are different. For a non-allday event this will show the time or the datetime if the event start and end date are different.

end-necessary-long Same as end-necessary but uses datelong and datetimelong.

By default all-day events have no times. To see a start and end time anyway simply add *-full* to the end of any template with start/end, for instance *start-time* becomes *start-time-full* and will always show start and end times (instead of being empty for all-day events).

In addition there are colors: *black*, *red*, *green*, *yellow*, *blue*, *magenta*, *cyan*, *white* (and their bold versions: *red-bold*, etc.). There is also *reset*, which clears the styling, and *bold*, which is the normal bold.

For example the below command with print the title and description of all events today.

```
khal list --format "{title} {description}"
```

--day-format DAYFORMAT
works similar to `--format`, but for day headings. It only has a small number of options (in addition to all of the color options):

date The date in dateformat.

date-long The date in longdateformat.

name The date's name (*Monday*, *Tuesday*,...) or *today* or *tomorrow*.

If the *–day-format* is passed an empty string then it will not print the day headers (for an empty line pass in a whitespace character).

2.3.2 dates

Almost everywhere khal accepts dates, khal should recognize relative date names like *today*, *tomorrow* and the names of the days of the week (also in three letters abbreviated form). Week day names get interpreted as the date of the next occurrence of a day with that name. The name of the current day gets interpreted as that date *next* week (i.e. seven days from now).

2.3.3 Commands

list

shows all events scheduled for a given date (or datetime) range, with custom formatting:

```
khal list [-a CALENDAR ... | -d CALENDAR ...] [--format FORMAT]
[--day-format DAYFORMAT] [--once] [--notstarted] [START [END | DELTA] ]
```

START and END can both be given as dates, datetimes or times (it is assumed today is meant in the case of only a given time) in the formats configured in the configuration file. If END is not given, midnight of the start date is assumed. Today is used for START if it is not explicitly given. If DELTA, a (date)time range in the format *I{m,h,d}*, where *I* is an integer and *m* means minutes, *h* means hours, and *d* means days, is given, END is assumed to be START + DELTA. A value of *eod* is also accepted as DELTA and means the end of day of the start date. In addition the DELTA *week* may be used to specify that the daterange should actually be the week containing the START.

The *–once* option only allows events to appear once even if they are on multiple days. With the *–notstarted* option only events are shown that start after *START*.

at

shows all events scheduled for a given datetime. `khal at` should be supplied with a date and time, a time (the date is then assumed to be today) or the string *now*. `at` defaults to *now*. The `at` command works just like the `list` command, except it has an implicit end time of zero minutes after the start.

```
khal list [-a CALENDAR ... | -d CALENDAR ...] [--format FORMAT]
[--notstarted] [START DATE]
```

calendar

shows a calendar (similar to `cal(1)`) and list. `khal calendar` should understand the following syntax:

```
khal calendar [-a CALENDAR ... | -d CALENDAR ...] [START DATETIME]
[END DATETIME]
```

Date selection works exactly as for `khal list`. The displayed calendar contains three consecutive months, where the first month is the month containing the first given date. If today is included, it is highlighted. Have a look at `khal list` for a description of the options.

configure

will help users creating an initial configuration file. **configure** will refuse to run if there already is a configuration file.

import

lets the user import `.ics` files with the following syntax:

```
khal import [-a CALENDAR] [--batch] [--random-uid|-r] ICSFILE
```

If an event with the same UID is already present in the (implicitly) selected calendar `khal import` will ask before updating (i.e. overwriting) that old event with the imported one, unless –batch is given, than it will always update. If this behaviour is not desired, use the *–random-uid* flag to generate a new, random UID. If no calendar is specified (and not *–batch*), you will be asked to choose a calendar. You can either enter the number printed behind each calendar's name or any unique prefix of a calendar's name.

interactive

invokes the interactive version of khal, can also be invoked by calling **ikhal**. While ikhal can be used entirely with the keyboard, some elements respond if clicked on with a mouse (mostly by being selected).

When the calendar on the left is in focus, you can

- move through the calendar (default keybindings are the arrow keys, `space` and `backspace`, those keybindings are configurable in the config file)
- focus on the right column by pressing `tab` or `enter`
- re-focus on the current date, default keybinding `t` as in today
- marking a date range, default keybinding `v`, as in visual, think visual mode in Vim, pressing `esc` escape this visual mode
- if in visual mode, you can select the other end of the currently marked range, default keybinding `o` as in other (again as in Vim)
- create a new event on the currently focused day (or date range if a range is selected), default keybinding `n` as in new
- search for events, default keybinding `/`, a pop-up will ask for your search term

When an event list is in focus, you can

- view an event's details with pressing `enter` (or `tab`) and edit it with pressing `enter` (or `tab`) again (if `[default] event_view_always_visible` is set to True, the event in focus will always be shown in detail)
- toggle an event's deletion status, default keybinding `d` as in delete, events marked for deletion will appear with a `D` in front and will be deleted when khal exits.
- duplicate the selected event, default keybinding `p` as in duplicate (d was already taken)
- export the selected event, default keybinding `e`

In the event editor, you can

- jump to the next (previous) selectable element with pressing `tab` (`shift+tab`)

- quick save, default keybinding `meta+enter` (`meta` will probably be `alt`)

- use some common editing short cuts in most text fields (`ctrl+w` deletes word before cursor, `ctrl+u` (`ctrl+k`) deletes till the beginning (end) of the line, `ctrl+a` (`ctrl+e`) will jump to the beginning (end) of the line

- in the date and time field you can increment and decrement the number under the cursor with `ctrl+a` and `ctrl+x` (time in 15 minute steps)

- activate actions by pressing `enter` on text enclosed by angled brackets, e.g. *< Save >* (sometimes this might open a pop up)

Pressing `esc` will cancel the current action and/or take you back to the previously shown pane (i.e. what you see when you open ikhal), if you are at the start pane, ikhal will quit on pressing `esc` again.

new

allows for adding new events. `khal new` should understand the following syntax:

```
khal new [-a CALENDAR] [OPTIONS] [START [END | DELTA] [TIMEZONE] SUMMARY
[:: DESCRIPTION]]
```

where start- and enddatetime are either datetimes, times, or keywords and times in the formats defined in the config file. If no calendar is given via *-a*, the default calendar is used. **new** does not support *-d* and also *-a* may only be used once.

new accepts these combinations for start and endtimes (specifying the end is always optional):

- *datetime [datetime\time] [timezone]*

- *time [time] [timezone]*

- *date [date]*

where the formats for datetime and time are as follows:

- *datetime = (longdatetimeformat\datetimeformat\keyword-date timeformat)*

- *time = timeformat*

- *date = (longdateformat\dateformat)*

and *timezone*, which describes the timezone the events start and end time are in, should be a valid Olson DB identifier (like *Europe/Berlin* or *America/New_York*. If no timezone is given, the *defaulttimezone* as configured in the configuration file is used instead.

The exact format of longdatetimeformat, datetimeformat, timeformat, longdateformat and dateformat can be configured in the configuration file. Valid keywords for dates are *today*, *tomorrow*, the English name of all seven weekdays and their three letter abbreviations (their next occurrence is used).

If no end is given, the default length of one hour or one day (for all-day events) is used. If only a start time is given the new event is assumed to be starting today. If only a time is given for the event to end on, the event ends on the same day it starts on, unless that would make the event end before it has started, then the next day is used as end date

If a 24:00 time is configured (timeformat = %H:%M) an end time of *24:00* is accepted as the end of a given date.

If the **summary** contains the string *::*, everything after *::* is taken as the **description** of the new event, i.e., the "body" of the event (and *::* will be removed).

Passing the option `--interactive` (`-i`) makes all arguments optional and interactively prompts for required fields, then the event may be edited, the same way as in the *edit* command.

Options

- **-l, –location=LOCATION** specify where this event will be held.

- **-g, –categories=CATEGORIES** specify which categories this event belongs to. Comma separated list of categories. Beware: some servers (e.g. SOGo) do not support multiple categories.

- **-r, –repeat=RRULE** specify if and how this event should be recurring. Valid values for *RRULE* are *daily*, *weekly*, *monthly* and *yearly*

- **-u, –until=UNTIL** specify until when a recurring event should run

- **–alarm DURATION** will add an alarm DURATION before the start of the event, *DURATION* should look like *1day 10minutes* or *1d3H10m*, negative *DURATIONs* will set alarm after the start of the event.

Examples

```
khal new 18:00 Awesome Event
```

adds a new event starting today at 18:00 with summary 'awesome event' (lasting for the default time of one hour) to the default calendar

```
khal new tomorrow 16:30 Coffee Break
```

adds a new event tomorrow at 16:30

```
khal new 25.10. 18:00 24:00 Another Event :: with Alice and Bob
```

adds a new event on 25th of October lasting from 18:00 to 24:00 with an additional description

```
khal new -a work 26.07. Great Event -g meeting -r weekly
```

adds a new all day event on 26th of July to the calendar *work* which recurs every week.

edit

an interactive command for editing and deleting events using a search string

```
khal edit [--show-past] event_search_string
```

the command will loop through all events that match the search string, prompting the user to delete, or change attributes.

printcalendars

prints a list of all configured calendars.

printformats

prints a fixed date (*2013-12-21 10:09*) in all configured date(time) formats. This is supposed to help check if those formats are configured as intended.

search

search for events matching a search string and print them. Currently recurring events are only printed once. No advanced search features are currently supported.

The command

```
khal search party
```

prints all events matching *party*.

2.4 Standards

khal tries to follow standards and RFCs (most importantly **RFC 5545** *iCalendar*) wherever possible. Known intentional and unintentional deviations are listed below.

2.4.1 RDATE;VALUE=PERIOD

RDATE s with *PERIOD* values are currently not supported, as icalendar does does not support it yet. Please submit any real world examples of events with *RDATE;VALUE=PERIOD* you might encounter (khal will print warnings if you have any in your calendars).

2.4.2 RANGE=THISANDPRIOR

Recurrent events with the *RANGE=THISANDPRIOR* are and will not be [1] supported by khal, as applications supporting the latest standard MUST NOT create those. khal will print a warning if it encounters an event containing *RANGE=THISANDPRIOR*.

2.4.3 Events with neither END nor DURATION

While the RFC states:

```
A calendar entry with a "DTSTART" property but no "DTEND"
property does not take up any time. It is intended to represent
an event that is associated with a given calendar date and time
of day, such as an anniversary. Since the event does not take up
any time, it MUST NOT be used to record busy time no matter what
the value for the "TRANSP" property.
```

khal transforms those events into all-day events lasting for one day (the start date). As long a those events do not get edited, these changes will not be written to the vdir (and with that to the CalDAV server). Any timezone information that was associated with the start date gets discarded.

Note: While the main rationale for this behaviour was laziness on part of khal's main author, other calendar software shows the same behaviour (e.g. Google Calendar and Evolution).

---

[1] unless a lot of users request this feature

2.4.4 Timezones

Getting localized time right, seems to be the most difficult part about calendaring (and messing it up ends in missing the one important meeting of the week). So I'll briefly describe here, how khal tries to handle timezone information, which information it can handle and which it can't.

In general, there are two different type of events. *Localized events* (with *localized* start and end datetimes) which have timezone information attached to their start and end datetimes, and *floating* events (with *floating* start and end datetimes), which have no timezone information attached (all-day events, events that last for complete days are floating as well). Localized events are always observed at the same UTC (no matter what time zone the observer is in), but different local times. On the other hand, floating events are always observed at the same local time, which might be different in UTC.

In khal all localized datetimes are saved to the local database as UTC. Datetimes that are already UTC, e.g. 19980119T070000Z, are saved as such, others are converted to UTC (but don't worry, the timezone information does not get lost). Floating events get saved in floating time, independently of the localized events.

If you want to look up which events take place at a specified datetime, khal always expects that you want to know what events take place at that *local* datetime. Therefore, the (local) datetime you asked for gets converted to UTC, the appropriate *localized* events get selected and presented with their start and end datetimes *converted* to *your local datetime*. For floating events no conversion is necessary.

Khal (i.e. icalendar) can understand all timezone identifiers as used in the Olson DB and custom timezone definitions, if those VTIMEZONE components are placed before the VEVENTS that make use of them (as most calendar programs seem to do). In case a unknown (or unsupported) timezone is found, khal will assume you want that event to be placed in the *default timezone* (which can be configured in the configuration file as well).

khal expects you *always* want *all* start and end datetimes displayed in *local time* (which can be set in the configuration file as well, otherwise your computer's timezone is used).

2.5 Feedback

Note: All participants must follow the pimutils Code of Conduct.

Please do provide feedback if *khal* works for you or even more importantly, if it doesn't. Feature requests and other ideas on how to improve khal are also welcome (see below).

In case you are not satisfied with khal, there are at least two other projects with similar aims you might want to check out: calendar-cli (no offline storage and a bit different scope) and gcalcli (only works with google's calendar).

2.5.1 Submitting a Bug

If you found a bug or any part of khal isn't working as you expected, please check if that bug is also present in the latest version from github (see Installation) and is not already reported (you still might want to comment on an already open issue).

If it isn't, please open a new bug. In case you submit a new bug please include:

- how you ran khal (please run in verbose mode with *-v*)

- what you expected khal to do

- what it did instead

- everything khal printed to the screen (you may redact private details)

- in case khal complains about a specific .ics file, please include that as well (or create a .ics which leads to the same error without any private information)

- the version of khal and python you are using, which operating system you are using and how you installed khal

2.5.2 Suggesting Features

If you believe khal is lacking a useful feature or some part of khal is not working the way you think it should, please first check if there isn't already a relevant issue for it and otherwise open a new one.

2.5.3 Contact

- You might get quick answers on the #pimutils IRC channel on Freenode, if nobody is answering you, please hang around for a bit. You can also use this channel for general discussions around **khal** and related tools.

- Open a github issue

- If the above mentioned methods do not work, you can always contact the main developer.

2.6 Hacking

Note: All participants must follow the pimutils Code of Conduct.

Please discuss your ideas with us, before investing a lot of time into khal (to make sure, no efforts are wasted). Also if you have any questions on khal's codebase, please don't hesitate to *contact* us, we will gladly provide you with any information you need or set up a joined hacking session.

The preferred way of submitting patches is via github pull requests (PRs). If you are not comfortable with that, please *contact* us and we can work out something else. If you have something working, don't hesitate to open a PR very early and ask for opinions.

Before we will accept your PR, we will ask you to:

- add yourself to `AUTHORS.txt` if you haven't done it before

- add a note to `CHANGELOG.rst` explaining your changes (if you changed anything user facing)

- edit the documentation (again, only if your changes impact the way users interact with khal)

- make sure all tests pass (see below)

- write some tests covering your patch (this really is mandatory, unless it's in the urwid part, testing which is often difficult)

- make sure your patch conforms with **PEP 008** (should be covered by passing tests)

2.6.1 General notes for developing khal (and lots of other python packages)

The below notes are meant to be helpful if you are new to developing python packages in general and/or khal specifically. While some of these notes are therefore specific to khal, most should apply to lots of other python packages developed in comparable setup. Please note that all commands (if not otherwise noted) should be executed at the root of khal's source directory, i.e., the directory you got by cloning khal via git.

Please note that fixes and enhancements to these notes are very welcome, too.

Isolation

When working on khal (or for any other python package) it has proved very beneficial to create a new *virtual environ-ments* (with the help of virtualenv), to be able to run khal in isolation from your globally installed python packages and to ensure to not run into any conflicts very different python packages depend on different version of the same library. virtualenvwrapper (for bash and zsh users) and virtualfish (for fish users) are handy wrappers that make working with virtual environments very comfortable.

After you have created and activated a virtual environment, it is recommend to install khal via **pip install -e .** (from the base of khal's source directory), this install khal in an editable development mode, where you do not have to reinstall khal after every change you made, but where khal will always have picked up all the latest changes (with the exception of adding new files, hereafter reinstall khal *is* necessary).

Testing

khal has an extensive self test suite, that lives in `tests/`. To run the test suite, install *pytest* and run **py.test tests**, pytest will then collect and run all tests and report on any failures (which you should then proceed to fix). If you only want to run tests contained in one file, run, e.g., **py.test tests/backend_test.py**. If you only want to run one or more specific tests, you can filter for them with **py.test -k calendar**, which would only run tests including *calendar* in their name.

To ensure that khal runs on all currently supported version of python, the self test suite should also be run with all supported versions of python. This can locally be done with tox. After installing tox, running tox will create new virtual environments (which it will reuse on later runs), one for each python version specified in `tox.ini`, run the test suite and report on it.

If you open a pull request (*PR*) on github, the continuous integration service travis CI will automatically perform exactly those tasks and then comment on the success or failure.

If you make any non-trivial changes to khal, please ensure that those changes are covered by (new) tests. As testing **ikhal** (the part of **khal** making use of urwid) has proven rather complicated (as can be seen in the lack tests covering that part of khal), automated testing of changes of that part is therefore not mandatory, but very welcome nonetheless.

To make sure all major code paths are run through at least once, please check the *coverage* the tests provide. This can be done with pytest-cov. After installing pytest-cov, running **py.test --cov khal --cov-report=html tests** will generate a html-based report on test coverage (which can be found in `htmlcov`), including a color-coded version of khal's source code, indicating which lines have been run and which haven't.

Debugging

For an improved debugging experience on the command line, pdb++ is recommended (install with **pip install pdbpp**. **pdb++** is a drop in replacement for python's default debugger, and can therefore be used like the default debugger, e.g., invoked by placing `import pdb; pdb.set_trace()` at the respective place. One of the main reasons for choosing **pdb++** over alternatives like IPython's debugger ipdb, is that it works nicely with **pytest**, e.g., running *py.test –pdb tests* will drop you at a **pdb++** prompt at the place of the first failing test.

Documentation

Khal's documentation, which is living in `doc`, is using sphinx to generate the html documentation as well as the man page from the same sources. After install *sphinx* and *sphinxcontrib-newsfeed* you should be able to build the documentation with **make html** and **make man** respectively from the root of the `doc` directory (note that this requires *GNU make*, so on some system running **gmake** make be required).

If you make any changes to how a user would interact with khal, please change or add the relevant section(s) in the documentation, which uses the reStructuredText format, which shouldn't be to hard after looking at some of the existing documentation (even for users who never used it before).

Note: The file `doc/source/configspec.rst` is auto-generated on making the documentation from the file `khal/settings/khal.spec`. So instead of editing the former, please edit the later, run make and include both changes in your patch.

Also, summarize your changes in `CHANGELOG.rst`, pointing readers to the (updated) documentation is fine.

Code Style

khal's source code should adhere to the rules laid out in **PEP 008**, with the exception of allowing line lengths of up to 100 characters if it improves overall legibility (use your judgement). This can be checked by installing and running flake8 (run with **flake8** from khal's source directory), which will also be run with tox and travisCI, see section above.

We try to document the parameters functions and methods accept, including their types, and their return values in the sphinx style, though this is currently not used thoroughly.

Note that we try to use double quotes for human readable strings, e.g., strings that one would internationalize and single quotes for strings used as identifiers, e.g., in dictionary keys:

```
my_event['greeting'] = "Hello World!"
```

2.6.2 iCalendar peculiarities

These notes are meant for people who want to deep dive into `khal.khalendar.backend.py` and are not recommended reading material for anyone else.

A single *.ics* can contain several VEVENTS, which might or might not be the part of the same event. This can lead to issues with straight forward implementations. Some of these, and the way khal is dealing with them, are described below.

While one would expect every VEVENT to have its own unique UID (for what it's worth they are named *unique identifier*), there is a case where several VEVENTS have the same UID, but do describe the same (recurring) event. In this case, one VEVENT, containing an RRULE or RDATE element would be the *proto* event, from which all recurrence instances are derived. All other VEVENTS with the same UID would then have a RECURRENCE-ID element (I'll call them *child* event from now on) and describe deviations of at least one recurrence instance (RECURRENCE-ID elements can also have the added property RANGE=THISANDFUTURE, meaning the deviations described by this child event also apply to all further recurrence instances.

Because it is possible that an event already in the database consists of a master event and at least one child event gets updated and than consists only of a master event, we currently *delete* all events with the same UID from the database when inserting or updating a new event. But this means that we need to update an event always at once (master and all child events) at the same time (using *Calendar.update()* or *Calendar.new()* in this case)

As this wouldn't be bad enough, the standard looses no words on the ordering on those VEVENTS in any given *.ics* file (at least I didn't find any). Not only can the proto event be *behind* any or all RECURRENCE-ID events, but also events with different UIDs can be in between.

We therefore currently first collect all events with the same UID and than sort those by their type (proto or child), and the children by the value of the RECURRENCE-ID property.

2.7 Changelog

All notable changes to this project should be documented here. For more detailed information have a look at the git log.

Package maintainers and users who have to manually update their installation may want to subscribe to GitHub's tag feed.

2.7.1 0.9.2

released 2017-02-13

- FIX if weekstart != 0 ikhal would show wrong weekday names
- FIX allday events added with *khal new DATE TIMEDELTA* (e.g., 2017-01-18 3d) were lasting one day too long
- FIX no more crashes when using timezones that have a constant UTC offset (like UTC itself)
- FIX updated outdated zsh completion file
- FIX display search results for events with neither DTEND nor DURATION
- FIX display search results that are all-day events
- in ikhal, update the date-titles on date change
- FIX printing a new event's path if [default] print_new = path
- FIX width of calendar in *khal calendar* was off by two if locale.weeknumbers was set to "right"
- CHANGED default *agenda_day_format* to include the actual date of the day
- NEW configuration option: [view]dynamic_days = True, if set to False, ikhal's right column behaves similar as it did in 0.8.x

2.7.2 0.9.1

released 2017-01-25

- FIX detecting not understood timezone information failed on python 3.6, this may lead to erroneous offsets in start and end times for those events, as those datetimes were treated as if they were in the system's local time, not as if they are in the (possibly) configured default_timezone.
- python 3.6 is now officially supported

2.7.3 0.9.0

released 2017-01-24

Dependency Changes

- vdirsyncer isn't a hard dependency any more

Bug Fixes

- fixed various bugs in *configure*

- fix bug in *new* that surfaces when date(time)format does contain a year

- fix bug in *import* that allows importing into read-only and/or non-default calendar

- fix how color discovered in calendars

Backwards Incompatibilities

- calendar path is now a glob without recursion for discover, if your calendars are no longer found, please consult the documentation (Taylor Money)

- *at* command now works like *list* with a timedelta of *0m*, this means that *at* will no longer print events that end at exactly the time asked for (Taylor Money)

- renamed *agenda* to *list* (Taylor Money)

- removed *days* configuration option in favor of *timedelta*, see documentation for details (Taylor Money)

- configuration file path $XDG_CONFIG_HOME/khal/config is now supported and $XDG_CONFIG_HOME/khal/khal.conf deprecated

- ikhal: introduction of three different new frame styles, new allowed values for *[view] frame* are *False*, *width*, *color*, *top* (with default *False*), *True* isn't allowed any more, please provide feedback over the usual channels if and which of those you consider useful as some of those might be removed in future releases (Christian Geier)

- removed configuration variable *encoding* (in section [locale]), the correct locale should now be figured out automatically (Markus Unterwaditzer)

- events that start and end at the same time are now displayed as if their duration was one hour instead of one day (Guilhem Saurel)

Enhancements

- (nearly) all commands allow formatting of how events are printed with *–format*, also see the new configuration options *event_format*, *agenda_event_format*, *agenda_day_format* (Taylor Money)

- support for categories (and add *-g* flag for *khal new*) (Pierre David)

- search results are now sorted by start date (Taylor Money)

- added command *edit*, which also allows deletion of events (Taylor Money)

- *new* has interactive option (Taylor Money)

- *import* can now import multiple files at once (Christian Geier)

ikhal

- BUGFIX no more crashing if invalid date is entered and mini-calendar displayed

- make keybinding for quitting configurable, defaults to *q* and *Q*, escape only backtracks to last pane but doesn't exit khal anymore (Christian Geier)

- default keybinding changed: *tab* no longer shows details of focused events and does not open the event editor either (Christian Geier)

- right column changed, it will now show as many days/events as fit, if users move to another date (while the event column is in focus), that date should be highlighted in the calendar (Christian Geier)

- cursor indicates which element is selected

2.7.4 0.8.4

released 2016-10-06

- **IMPORTANT BUGFIX** fixed a bug that lead to imported events being erroneously shifted if they had a time-zone identifier that wasn't an Olson database identifier. All users are advised to upgrade as soon as possible. To see if you are affected by this and how to resolve any issues, please see the release announcement (khal/doc/source/news/khal084.rst or http://lostpackets.de/khal/news/khal084.html). Thanks to Wayne Werner for finding and reporting this bug.

2.7.5 0.8.3

released 2016-08-28

- fixed some bugs in the test suite on different operating systems
- fixed a check for icalendar files containing RDATEs

2.7.6 0.8.2

released on 2016-05-16

- fixed some bugs in *configure* that would lead to invalid configuration files and crashes (Christian Geier)
- fixed detecting of icalendar version (Markus Unterwaditzer)

2.7.7 0.8.1

released on 2016-04-13

- fix bug in CalendarWidget.set_focus_date() (Christian Geier)

2.7.8 0.8.0

released on 2016-04-13

- BREAKING CHANGE: python 2 is no longer supported (Hugo Osvaldo Barrera)
- updated dependency: vdirsyncer >= 0.5.2
- make tests work with icalendar 3.9.2 (no functional changes) (Christian Geier)
- new dependency: freezegun (only for running the tests)
- khal's git repository moved to https://github.com/pimutils/khal
- support for showing the birthday of contacts with no FN property (Hugo Osvaldo Barrera)
- increased start up time when coloring is enabled (Christian Geier)
- improved color support (256 colors and 24-bit colors), see configuration documentation for details (Sebastian Hamann)

- renamed color *grey* to *gray* (Sebastian Hamann)

- in *khal new* treat 24:00 as the end of a day/00:00 of the next (Christian Geier)

- new allowed value for a calendar's color: *auto* (also the new default), if set, khal will try to read a file called *color* from that calendar's vdir (see vdirsyncer's documentation on *metasync*). If that file is not present or its contents is not understood, the default color will be used (Christian Geier)

- new allowed value for calendar's type: *discover*, if set, khal will (recursively) search that calendar's path for valid vdirs and add those to the configured calendars (Christian Geier)

- new command *configure* which should help new users set up a configuration file (Christian Geier)

- warn user when parsing broken icalendar files, this requires icalendar > 3.9.2 (Christian Geier)

- khal will now strip all ANSI escape codes when it detects that stdout is no tty, this behaviour can be overwritten with the new options –color/ –no-color (Markus Unterwaditzer)

- calendar and agenda have a new option –week, if set all events from current week (or the week containing the given date) are shown (Stephan Weller)

- new option –alarm DURATION for *new* (Max Voit)

ikhal

- basic export of events from event editor pane and from event lists (default keybinding: *e* (Filip Pytloun)

- pressing *enter* in a date editing widget will now open a small calendar widget, arrow keys can be used to select a date, enter (or escape) will close it again (Christian Geier)

- in highlight/date range selection mode the other end can be selected, default keybinding *o* (as in *Other*) (Christian Geier)

- basic search is now supported (default keybinding */*) (Christian Geier)

- in the event editor and pop-up Dialogs select the next (previous) item with tab (shift tab) (Christian Geier)

- only allow saving when starttime < endtime (Christian Geier)

- the event editor now allows editing of alarms (but khal will not actually alarm you at the given time) (Johannes Wienke)

2.7.9 0.7.0

released on 2015-11-24

There are no new or dropped dependencies.

- most of the internal representation of events was rewritten, the current benefit is that floating events are properly represented now, hopefully more is to come (Christian Geier)

- *printformats* uses a more sensible date now (John Shea)

- khal and ikhal can now highlight dates with events, at the moment, enabling it does noticably slow down (i)khal's start; set *[default] highlight_event_days = True* and see section *[highlight_days]* for further configuration (Dominik Joe Pantůček)

- fixed line wrapping for *at* (Thomas Schape)

- *calendar* and *agenda* optionally print location and description of all events, enable with the new –full/-f flag (Thomas Schaper)

- updated and improved zsh completion file (Oliver Kiddle)

- FIX: deleting events did not always work if an event with the same filename existed in an other calendar (but no data lost incurred) (Christian Geier)

ikhal

- events are now displayed nicer (Thomas Glanzmann)

- support for colorschemes, a *light* and *dark* one are currently included, help is wanted to make them prettier and more functional (config option *[view] theme: (dark|light)*) (Christian Geier)

- ikhal can now display frames around some user interface elements, making it nicer to look at in some eyes (config option *[view] frame: True*) (Christian Geier)

- events can now be duplicated (default keybinding: *p*) (Christian Geier)

- events created while time ranges are selected (default keybinding to enable date range selection: *v*) will default to that date range (Christian Geier)

- when trying to delete recurring events, users are now asked if they want to delete the complete event or just this instance (Christian Geier)

2.7.10 0.6.0

2015-07-15

- BUGFIX Recurrent events with a THISANDFUTURE parameter could affect other events. This could lead to events not being found by the normal lookup functionality when they should and being found when they shouldn't. As the second case should result in an error that nobody reported yet, I hope nobody got bitten by this.

- new dependency for running the tests: freezegun

- new dependency for setup from scm: setuptools_scm

- khal now needs to be installed for building the documentation

- ikhal's should now support ctrl-e, ctrl-a, ctrl-k and ctrl-u in editable text fields (Thomas Glanzmann)

- ikhal: space and backspace are new (additional) default keybindings for right and left (Pierre David)

- when editing descriptions you can now insert new lines (Thomas Glanzmann)

- khal should not choose an arbitrary default calendar anymore (Markus Unterwaditzer)

- the zsh completion file has been updated (Hugo Osvaldo Barrera)

- new command *import* lets users import .ics files (Christian Geier)

- khal should accept relative dates on the command line (today, tomorrow and weekday names) (Christian Geier)

- keybinding for saving an event from ikhal's event editor (default is *meta + enter*) (Christian Geier)

2.7.11 0.5.0

released on 2015-06-01

- fixed several bugs relating to events with unknown timezones but UNTIL, RDATE or EXDATE properties that are in Zulu time (thanks to Michele Baldessari for reporting those)

- bugfix: on systems with a local time of UTC-X dealing with allday events lead to crashes

- bugfix: British summer time is recognized as daylight saving time (Bradley Jones)
- compatibility with vdirsyncer 0.5
- new command *search* allows to search for events
- user changeable keybindings in ikhal, with hjkl as default alternatives for arrows in calendar browser, see documentation for more details
- new command *at* shows all events scheduled for a specific datetime
- support for reading birthdays from vcard collections (set calendar/collection *type* to *birthdays*)
- new command *printformats* prints a fixed date in all configured date-time settings
- *new* now supports the *–until/-u* flag to specify until when recurring events should run (Micah Nordland)
- python 3 (>= 3.3) support (Hugo Osvaldo Barrera)

ikhal

- minimal support for reccurring events in ikhal's editor (Micah Nordland)
- configurable view size in ikhal (Bradley Jones)
- show events organizers (Bradley Jones)
- major reorganisation of ikhal layout (Markus Unterwaditzer)

2.7.12 0.4.0

released on 2015-02-02

dependency changes

- new dependency: click>3.2
- removed dependency: docopt
- note to package mantainers: *requirements.txt* has been removed, dependencies are still listed in *setup.py*

note to users

- users will need to delete the local database, no data should be lost (and khal will inform the user about this)

new and changed features

- new config_option: *[default] print_new*, lets the user decide what should be printed after adding a new event
- new config option: *[default] show_all_days* lets users decide if they want to see days without any events in agenda and calendar view (thanks to Pierre David)
- khal (and ikhal) can now display weeknumbers
- khal new can now create repetitive events (with –repeat), see documentation (thanks to Eric Scheibler)
- config file: the debug option has been removed (use *khal -v* instead)

- FIX: vtimezones were not assembled properly, this lead to spurious offsets of events in some other calendar applications

- change in behaviour: recurring events are now always expanded until 2037

- major speedup in inserting events into the caching database, especially noticeable when running khal for the first time or after an deleting the database (Thanks to Markus Unterwaditzer)

- better support for broken events, e.g. events ending before they start (Thanks to Markus Unterwaditzer)

- more recurrence rules are supported, khal will print warnings on unsupported rules

ikhal

- ikhal's calendar should now be filled on startup

- pressing *t* refocuses on today

- pressing ctrl-w in input fields should delete the last word before the cursor

- when the focus is set on the events list/editor, the current date should still be visible in the calendar

2.7.13 0.3.1

released on 2014-09-08

- FIX: events deleted in the vdir are not shown anymore in khal. You might want to delete your local database file, if you have deleted any events on the server.

- FIX: in some cases non-ascii characters were printed even if unicode_symbols is set to False in the config

- FIX: events with different start and end timezones are now properly exported (the end timezone was disregarded when building an icalendar, but since timezones cannot be edited anyway, this shouldn't have caused any problems)

- FIX: calendars marked as read-only in the configuration file should now really be read-only

2.7.14 0.3.0

released on 2014-09-03

- **new unified documentation**

    - html documentation (website) and man pages are all generated from the same sources via sphinx (type *make html* or *make man* in doc/, the result will be build in *build/html* or *build/man* respectively

    - the new documentation lives in doc/

    - the package sphinxcontrib-newsfeed is needed for generating the html version (for generating an RSS feed)

    - the man pages live doc/build/man/, they can be build by running *make man* in doc/sphinx/

- new dependencies: configobj, tzlocal>=1.0

- **IMPORTANT**: the configuration file's syntax changed (again), have a look at the new documentation for details

- local_timezone and default_timezone will now be set to the timezone the computer is set to (if they are not set in the configuration file)

2.8 FAQ

Frequently asked questions:

- **start up of khal and ikhal is very slow** In some case the pytz (python timezone) is only available as a zip file, as pytz accesses several parts during initialization this takes some time. If *time python -c "import pytz; pytz.timezone('Europe/Berlin')"* takes nearly as much time as running khal, uncompressing that file via pytz via *(sudo) pip unzip pytz* might help.

- **ikhal raises an Exception: AttributeError: 'module' object has no attribute 'SimpleFocusListWalker'** You probably need to upgrade urwid to version 1.1.0, if your OS does come with an older version of *urwid* you can install the latest version to userspace (with out messing up your default installation) with *pip install –upgrade urwid –user*.

- **Installation stops with an error: source/str_util.c:25:20: fatal error: Python.h: No such file or directory** You do not have the Python development headers installed, on Debian based Distributions you can install them via *aptitude install python-dev*.

2.9 License

khal is released under the Expat/MIT License:

```
Copyright (c) 2013-2017 Christian Geier et al.

Permission is hereby granted, free of charge, to any person obtaining a copy of
this software and associated documentation files (the "Software"), to deal in
the Software without restriction, including without limitation the rights to
use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of
the Software, and to permit persons to whom the Software is furnished to do so,
subject to the following conditions:

The above copyright notice and this permission notice shall be included in all
copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR
IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS
FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR
COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER
IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN
CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.
```

2.10 News

Below is a list of new releases and other khal related news. This is also available as an rss feed .

2.10.1 khal v0.9.2 released

Published on *2017-02-13*

This is an **important bug fix release**, that fixes a bunch of different bugs, but most importantly:

- if weekstart != 0 ikhal would show wrong weekday names
- allday events added with *khal new DATE TIMEDELTA* (e.g., 2017-01-18 3d) were lasting one day too long

Special thanks to Tom Rushworth for finding and reporting both bugs!

All other fixed bugs would be rather obvious if you happened to run into them, as they would lead to khal crashing in one way or another.

One new feature made its way into this release as well, which is good news for all users pining for the way ikhal's right column behaved in pre 0.9.0 days: setting new configuration option [view]dynamic_days=False, will make that column behave similar as it used to.

> **Warning:** All users of khal 0.9.x are advised to **upgrade as soon as possible**.

Users of khal 0.8.x are not affected by either bug.

Get khal v0.9.2 from this site, or from pypi.

2.10.2 khal v0.9.1 released

Published on *2017-01-25*

This is a bug fix release for python 3.6.

Under python 3.6, datetimes with timezone information that is missing from the icalendar file would be treated if they were in the system's local timezone, not as if they were in khal's configured default timezone. This could therefore lead to erroneous offsets in start and end times for those events.

To check if you are affected by this bug, delete khal's database file (usually `~/.local/share/khal/khal.db`), rerun khal and watch for error messages that look like the one below:

> warning: DTSTART localized in invalid or incomprehensible timezone *FOO* in events/event_dt_local_missing_tz.ics. This could lead to this event being wrongly displayed.

All users (of python 3.6) are advised to upgrade as soon as possible.

Get khal v0.9.1 from this site, or from pypi.

2.10.3 khal v0.9.0 released

Published on *2017-01-24*

This is probably the biggest release of khal to date, that is, the one with the most changes since the last release. This changes are made up of a bunch of bug fixes and enhancements. Unfortunately, some of these break backwards compatibiliy, many of which will make themselves noticeable, because your config file will no longer be valid, consult the changelog or the documentation. Noticable is also, that command *agenda* has been renamed to *list*

Some of the larger changes include, among others, new configuration options on how events are printed (thanks to first time contributor Taylor Money) and a new look for ikhal's event-list column.

Have a look at the changelog for more complete list of new features (of which there are many).

Get khal v0.9.0 from this site, or from pypi.

2.10.4 khal v0.7.1 released

Published on *2016-10-11*

khal v0.7.1 (pypi) is a bugfix release that fixes a **critical bug** in *khal import*. This is a backport of the fix that got released with v0.8.4, for those users than cannot (or *really* don't want to) upgrade to a more recent version of khal (most likely because of the dropped support for python 2). Please note, that khal v0.7.x is generally *not maintained* anymore, will not receive any new features, and any non-critical bugs will not be fixed either.

See the 0.8.4 release announcement for more details regarding the fixed bug.

2.10.5 khal v0.8.4 released

Published on *2016-10-06*

khal v0.8.4 (pypi) is a bugfix release that fixes a **critical bug** in *khal import*. **All users are advised to upgrade as soon as possible**.

Details

If importing events from *.ics* files, any VTIMEZONEs (specifications of the timezone) would *not* be imported with those events. As khal understands Olson DB timezone specifiers (such as "Europe/Berlin" or "America/New_York", events using those timezones are displayed in the correct timezone, but all other events are displayed as if they were in the configured *default timezone*. **This can lead to imported events being shown at wrong times!**

Solution

First, please upgrade khal to either v0.8.4 or, if you are using a version of khal directly from the git repository, upgrade to the latest version from github.

To see if you are affected by this bug, delete your local khal caching db, (usually *~/.local/share/khal/khal.db*), re-run khal and watch out for lines looking like this: `warning: $PROPERTY has invalid or incomprehensible timezone information in $long_uid.ics in $my_collection.` You will then need to edit these files by hand and either replace the timezone identifiers with the corresponding one from the Olson DB (e.g., change *Europe_Berlin* to *Europe/Berlin*) or copy original VTIMZONE definition in.

If you have any problems with this, please either open an issue at github or come into our irc channel (*#pimutils* on Freenode).

We are sorry for any inconveniences this is causing you!

2.10.6 khal v0.8.3 released

Published on *2016-08-28*

khal v0.8.3 (pypi) is a maintenance release that fixes several bugs, mostly in the test suite. If khal is working fine for you, there is no need to upgrade.

2.10.7 khal v0.8.2 released

Published on *2016-05-16*

khal v0.8.2 (pypi) is a maintenance release that fixes several bugs in *configure* that would lead to crashes during the initial configuration and following runs of khal (due to an invalid configuration file getting written to disk) and improves the detection of the installed icalendar version.

If khal currently works for you, there is no need for an upgrade.

2.10.8 khal v0.8.1 released

Published on *2016-04-13*

The second version released today (khal v0.8.1, yes, also on pypi) fixes a bug in the CalendarWidget() that probably would not have trigged but made the tests fail.

2.10.9 khal v0.8.0 released

Published on *2016-04-13*

The latest version of khal has been released: khal v0.8.0 (as always, also on pypi).

We have recently dropped python 2 support, so this release is the first one that only supports python 3 (3.3+).

There is one more backwards incompatible change: The color *grey* has been renamed to *gray*, if you use it in your configuration file, you will need to update to *gray*.

There are some new features that should be configuring khal easier, especially for new users (e.g., new command *configure* helps with the initial configuration). Also alarms can now be entered either when creating new events with *new* or when editing them in ikhal.

Have a look at the changelog for more complete list of new features (of which there are many).

2.10.10 khal v0.7.0 released

Published on *2015-11-24*

The latest version of khal has been released: khal v0.7.0 (as always, also on pypi).

This release brings a lot of new features, by an ever increasing number of new contributors (welcome everyone!). With highlighting of days that have events we now have one of the most requested features implemented (because it does noticeably slow down khal's start it is disabled by default, startup performance will hopefully be increased soon). Among the other new features are duplicating events (in ikhal), prettier event display (also in ikhal) and a better zsh completion file.

Have a look at the changelog for more complete list of new features.

2.10.11 khal v0.6.0 released

Published on *2015-07-15*

Only six weeks after the last version khal v0.6.0 is now available (yes, also on pypi).

This release fixes an unfortunate bug which could lead to wrong shifts in other events when inserting complicated recurring events. All users are therefore advised to quickly upgrade to khal 0.6.

There are also quite a bunch of new features, among other nicer editing capabilities in ikhal's text edits and import of .ics files. For a more detailed list of changes, please have a look at the changelog (especially if you package khal).

2.10.12 khal v0.5.0 released

Published on *2015-06-01*

A new release of khal is here: khal v0.5.0 (also available on pypi).

This release brings a lot of new features (like rudimentary search support, user changeable keybindings in ikhal, new command *at*), python 3 support and some assorted bugfixes.

Thanks to everybody who contributed with bug reports, suggestions and code, especially to everyone contributing for the first time!

For a more detailed list of changes, please have a look at the changelog.

2.10.13 khal v0.4.0 released

Published on *2015-02-02*

A new release of khal is here: khal v0.4.0 (also available on pypi).

This release offers several functional improvements like better support for recurring events or a major speedup when creating the caching database and some new features like week number support or creating recurring events with *khal new –repeat*.

Note to users

khal now requires click instead of docopt and, as usual, the local database will need to be deleted.

For a more detailed list of changes, please have a look at the *Changelog*.

2.10.14 pycarddav and khal at 31c3

Published on *2014-12-09*

If you will be at 31C3 and would like to discuss the faults and merits of khal or pycarddav, commandline calendaring/addressbooking in general, your ideas or just have a beer or mate, I'd love to meet up. You can find my contact details under *Feedback*.

2.10.15 khal v0.3.1 released

Published on *2014-09-08*

A new release of khal is here: khal v0.3.1 (also available on pypi).

This is a bugfix release, bringing no new features. The last release suffered from a major bug, where events deleted on the server (and in the vdir) were not deleted in khal's caching database and therefore still displayed in khal. Therefore, after updating please delete your local database.

For more information on other fixed bugs, see *Changelog*.

2.10.16 khal v0.3 released

Published on *2014-09-03*

A new release of khal is here: khal v0.3.0 (also available on pypi).

If you want to update your installation from pypi, you can run *sudo pip install –upgrade khal*.

CHANGELOG

- **new unified documentation**
    - html documentation (website) and man pages are all generated from the same sources via sphinx (type *make html* or *make man* in doc/, the result will be build in *build/html* or *build/man* respectively (also available on Read the Docs)
    - the new documentation lives in doc/
    - the package sphinxcontrib-newsfeed is needed for generating the html version (for generating an RSS feed)
    - the man pages live doc/build/man/, they can be build by running *make man* in doc/sphinx/
- new dependencies: configobj, tzlocal>=1.0
- **IMPORTANT**: the configuration file's syntax changed (again), have a look at the new documentation for details
- local_timezone and default_timezone will now be set to the timezone the computer is set to (if they are not set in the configuration file)

2.10.17 khal v0.2 released

Published on *2014-06-27*

A new release of khal is here: khal v0.2.0 (also available on pypi).

If you want to update your installation from pypi, you can run *sudo pip install –upgrade khal*.

From now on *khal* relies on vdirsyncer for CalDAV sync. While this makes *khal* a bit more complicated to setup, *vdirsyncer* is much better tested than *khal* and also the bus factor increased (at least for parts of the project).

You might want to head over to the tutorial on how to setup *vdirsyncer*. Afterwards you will need to re-setup your *khal* configuration (copy the new example config file), also you will need to delete your old (local) database, so please make sure you did sync everything.

Also *khal*'s command line syntax changed qutie a bit, so you might want to head over the documentation.

2.10.18 khal v0.1.1 released

Published on *2014-05-07*

A small bugfix release: khal v0.1.0 Example config file now in source dist.

2.10.19 khal v0.1 released

Published on *2014-04-03*

The first release of khal is here: khal v0.1.0 (and also available on pypi now).

The next release, hopefully coming rather sooner than later, will get rid of its own CalDAV implementation, but instead use vdirsyncer; you can already try it out via checking out the branch *vdir* at github.

2.10.20 pycarddav and khal at 30c3

Published on *2013-12-13*

If you will be [30C3](#) and would like to discuss the faults and merits of khal or pycarddav, commandline calendaring/addressbooking in general, your ideas or just have a beer or mate, I'd love to meet up. You can find my contact details under *Feedback*.

2.10.21 Call for Testing

Published on *2013-11-19*

While there isn't a release yet, *khal* is, at least partly, in a usable shape by now. Please report any errors you stumble upon and improvement suggestions you have either via email or [github](#) (if you don't have any privacy concerns etc. I'd prefer you use github since it is public, but I'll soon set up a mailing list). [TODO.rst](#) gives you an idea about the plans I currently have for *khal*'s near future.

2.10.22 khal v0.9.2 released

Published on *2017-02-13*

This is an **important bug fix release**, that fixes a bunch of different bugs, but most importantly:

- if weekstart != 0 ikhal would show wrong weekday names
- allday events added with *khal new DATE TIMEDELTA* (e.g., 2017-01-18 3d) were lasting one day too long

Special thanks to Tom Rushworth for finding and reporting both bugs!

All other fixed bugs would be rather obvious if you happened to run into them, as they would lead to khal crashing in one way or another.

One new feature made its way into this release as well, which is good news for all users pining for the way ikhal's right column behaved in pre 0.9.0 days: setting new configuration option [view]dynamic_days=False, will make that column behave similar as it used to.

> **Warning:** All users of khal 0.9.x are advised to **upgrade as soon as possible**.

Users of khal 0.8.x are not affected by either bug.

Get [khal v0.9.2](#) from this site, or from [pypi](#).

2.10.23 khal v0.9.1 released

Published on *2017-01-25*

This is a bug fix release for python 3.6.

Under python 3.6, datetimes with timezone information that is missing from the icalendar file would be treated if they were in the system's local timezone, not as if they were in khal's configured default timezone. This could therefore lead to erroneous offsets in start and end times for those events.

To check if you are affected by this bug, delete khal's database file (usually `~/.local/share/khal/khal.db`), rerun khal and watch for error messages that look like the one below:

> warning: DTSTART localized in invalid or incomprehensible timezone *FOO* in events/event_dt_local_missing_tz.ics. This could lead to this event being wrongly displayed.

All users (of python 3.6) are advised to upgrade as soon as possible.

Get khal v0.9.1 from this site, or from pypi.

2.10.24 khal v0.9.0 released

Published on *2017-01-24*

This is probably the biggest release of khal to date, that is, the one with the most changes since the last release. This changes are made up of a bunch of bug fixes and enhancements. Unfortunately, some of these break backwards compatibiliy, many of which will make themselves noticeable, because your config file will no longer be valid, consult the changelog or the documentation. Noticable is also, that command *agenda* has been renamed to *list*

Some of the larger changes include, among others, new configuration options on how events are printed (thanks to first time contributor Taylor Money) and a new look for ikhal's event-list column.

Have a look at the changelog for more complete list of new features (of which there are many).

Get khal v0.9.0 from this site, or from pypi.

2.10.25 khal v0.7.1 released

Published on *2016-10-11*

khal v0.7.1 (pypi) is a bugfix release that fixes a **critical bug** in *khal import*. This is a backport of the fix that got released with *v0.8.4*, for those users than cannot (or *really* don't want to) upgrade to a more recent version of khal (most likely because of the dropped support for python 2). Please note, that khal v0.7.x is generally *not maintained* anymore, will not receive any new features, and any non-critical bugs will not be fixed either.

See the 0.8.4 release announcement for more details regarding the fixed bug.

2.10.26 khal v0.8.4 released

Published on *2016-10-06*

khal v0.8.4 (pypi) is a bugfix release that fixes a **critical bug** in *khal import*. **All users are advised to upgrade as soon as possible**.

Details

If importing events from *.ics* files, any VTIMEZONEs (specifications of the timezone) would *not* be imported with those events. As khal understands Olson DB timezone specifiers (such as "Europe/Berlin" or "America/New_York", events using those timezones are displayed in the correct timezone, but all other events are displayed as if they were in the configured *default timezone*. **This can lead to imported events being shown at wrong times!**

Solution

First, please upgrade khal to either v0.8.4 or, if you are using a version of khal directly from the git repository, upgrade to the latest version from github.

To see if you are affected by this bug, delete your local khal caching db, (usually *~/.local/share/khal/khal.db*), re-run khal and watch out for lines looking like this: `warning: $PROPERTY has invalid or incomprehensible timezone information in $long_uid.ics in $my_collection.` You

will then need to edit these files by hand and either replace the timezone identifiers with the corresponding one from the Olson DB (e.g., change *Europe_Berlin* to *Europe/Berlin*) or copy original VTIMZONE definition in.

If you have any problems with this, please either open an issue at github or come into our irc channel (*#pimutils* on Freenode).

We are sorry for any inconveniences this is causing you!

2.10.27 khal v0.8.3 released

Published on *2016-08-28*

khal v0.8.3 (pypi) is a maintenance release that fixes several bugs, mostly in the test suite. If khal is working fine for you, there is no need to upgrade.

2.10.28 khal v0.8.2 released

Published on *2016-05-16*

khal v0.8.2 (pypi) is a maintenance release that fixes several bugs in *configure* that would lead to crashes during the initial configuration and following runs of khal (due to an invalid configuration file getting written to disk) and improves the detection of the installed icalendar version.

If khal currently works for you, there is no need for an upgrade.

2.10.29 khal v0.8.1 released

Published on *2016-04-13*

The second version released today (khal v0.8.1, yes, also on pypi) fixes a bug in the CalendarWidget() that probably would not have trigged but made the tests fail.

2.10.30 khal v0.8.0 released

Published on *2016-04-13*

The latest version of khal has been released: khal v0.8.0 (as always, also on pypi).

We have recently dropped python 2 support, so this release is the first one that only supports python 3 (3.3+).

There is one more backwards incompatible change: The color *grey* has been renamed to *gray*, if you use it in your configuration file, you will need to update to *gray*.

There are some new features that should be configuring khal easier, especially for new users (e.g., new command *configure* helps with the initial configuration). Also alarms can now be entered either when creating new events with *new* or when editing them in ikhal.

Have a look at the changelog for more complete list of new features (of which there are many).

2.10.31 khal v0.7.0 released

Published on *2015-11-24*

The latest version of khal has been released: khal v0.7.0 (as always, also on pypi).

This release brings a lot of new features, by an ever increasing number of new contributors (welcome everyone!). With highlighting of days that have events we now have one of the most requested features implemented (because it does noticeably slow down khal's start it is disabled by default, startup performance will hopefully be increased soon). Among the other new features are duplicating events (in ikhal), prettier event display (also in ikhal) and a better zsh completion file.

Have a look at the changelog for more complete list of new features.

2.10.32 khal v0.6.0 released

Published on *2015-07-15*

Only six weeks after the last version khal v0.6.0 is now available (yes, also on pypi).

This release fixes an unfortunate bug which could lead to wrong shifts in other events when inserting complicated recurring events. All users are therefore advised to quickly upgrade to khal 0.6.

There are also quite a bunch of new features, among other nicer editing capabilities in ikhal's text edits and import of .ics files. For a more detailed list of changes, please have a look at the changelog (especially if you package khal).

2.10.33 khal v0.5.0 released

Published on *2015-06-01*

A new release of khal is here: khal v0.5.0 (also available on pypi).

This release brings a lot of new features (like rudimentary search support, user changeable keybindings in ikhal, new command *at*), python 3 support and some assorted bugfixes.

Thanks to everybody who contributed with bug reports, suggestions and code, especially to everyone contributing for the first time!

For a more detailed list of changes, please have a look at the changelog.

2.10.34 khal v0.4.0 released

Published on *2015-02-02*

A new release of khal is here: khal v0.4.0 (also available on pypi).

This release offers several functional improvements like better support for recurring events or a major speedup when creating the caching database and some new features like week number support or creating recurring events with *khal new –repeat*.

Note to users

khal now requires click instead of docopt and, as usual, the local database will need to be deleted.

For a more detailed list of changes, please have a look at the *Changelog*.

2.10.35 pycarddav and khal at 31c3

Published on *2014-12-09*

If you will be at 31C3 and would like to discuss the faults and merits of khal or pycarddav, commandline calendaring/addressbooking in general, your ideas or just have a beer or mate, I'd love to meet up. You can find my contact details under *Feedback*.

2.10.36 khal v0.3.1 released

Published on *2014-09-08*

A new release of khal is here: khal v0.3.1 (also available on pypi).

This is a bugfix release, bringing no new features. The last release suffered from a major bug, where events deleted on the server (and in the vdir) were not deleted in khal's caching database and therefore still displayed in khal. Therefore, after updating please delete your local database.

For more information on other fixed bugs, see *Changelog*.

2.10.37 khal v0.3 released

Published on *2014-09-03*

A new release of khal is here: khal v0.3.0 (also available on pypi).

If you want to update your installation from pypi, you can run *sudo pip install –upgrade khal*.

CHANGELOG

- **new unified documentation**
    - html documentation (website) and man pages are all generated from the same sources via sphinx (type *make html* or *make man* in doc/, the result will be build in *build/html* or *build/man* respectively (also available on Read the Docs)
    - the new documentation lives in doc/
    - the package sphinxcontrib-newsfeed is needed for generating the html version (for generating an RSS feed)
    - the man pages live doc/build/man/, they can be build by running *make man* in doc/sphinx/
- new dependencies: configobj, tzlocal>=1.0
- **IMPORTANT**: the configuration file's syntax changed (again), have a look at the new documentation for details
- local_timezone and default_timezone will now be set to the timezone the computer is set to (if they are not set in the configuration file)

2.10.38 khal v0.2 released

Published on *2014-06-27*

A new release of khal is here: khal v0.2.0 (also available on pypi).

If you want to update your installation from pypi, you can run *sudo pip install –upgrade khal*.

From now on *khal* relies on vdirsyncer for CalDAV sync. While this makes *khal* a bit more complicated to setup, *vdirsyncer* is much better tested than *khal* and also the bus factor increased (at least for parts of the project).

You might want to head over to the tutorial on how to setup *vdirsyncer*. Afterwards you will need to re-setup your *khal* configuration (copy the new example config file), also you will need to delete your old (local) database, so please make sure you did sync everything.

Also *khal*'s command line syntax changed qutie a bit, so you might want to head over the documentation.

2.10.39 khal v0.1.1 released

Published on *2014-05-07*

A small bugfix release: khal v0.1.0 Example config file now in source dist.

2.10.40 khal v0.1 released

Published on *2014-04-03*

The first release of khal is here: khal v0.1.0 (and also available on pypi now).

The next release, hopefully coming rather sooner than later, will get rid of its own CalDAV implementation, but instead use vdirsyncer; you can already try it out via checking out the branch *vdir* at github.

2.10.41 pycarddav and khal at 30c3

Published on *2013-12-13*

If you will be 30C3 and would like to discuss the faults and merits of khal or pycarddav, commandline calendaring/addressbooking in general, your ideas or just have a beer or mate, I'd love to meet up. You can find my contact details under *Feedback*.

2.10.42 Call for Testing

Published on *2013-11-19*

While there isn't a release yet, *khal* is, at least partly, in a usable shape by now. Please report any errors you stumble upon and improvement suggestions you have either via email or github (if you don't have any privacy concerns etc. I'd prefer you use github since it is public, but I'll soon set up a mailing list). TODO.rst gives you an idea about the plans I currently have for *khal*'s near future.

Symbols

–color/–no-color
    command line option, 14
–day-format DAYFORMAT
    command line option, 15
–format FORMAT
    command line option, 14
–version
    command line option, 14
-a CALENDAR
    command line option, 14
-c CONFIGFILE
    command line option, 14
-d CALENDAR
    command line option, 14
-h, –help
    command line option, 14
-v
    command line option, 14

C

command line option
    –color/–no-color, 14
    –day-format DAYFORMAT, 15
    –format FORMAT, 14
    –version, 14
    -a CALENDAR, 14
    -c CONFIGFILE, 14
    -d CALENDAR, 14
    -h, –help, 14
    -v, 14

P

Python Enhancement Proposals
    PEP 008, 22, 24

R

RFC
    RFC 5545, 20